

Towards Animated Visualization of Actors and Actions in a Learning Environment

Oleksandr Kolomiyets and Marie-Francine Moens

Department of Computer Science KU Leuven
`{oleksandr.kolomiyets,sien.moens}@cs.kuleuven.be`

Abstract. This paper describes ongoing research focused on natural language understanding and visualization of actors and actions extracted from narrative text. The technique employs a natural language processing pipeline for sophisticated syntactic and semantic analysis of text, and extracts information about events, actors and their roles in events, as well as temporal ordering of the events and spatial roles. This kind of information is traditionally considered indicative for text comprehension skill tests with novel readers. The visualization is implemented in the 3D graphics prototyping environment Alice, which provides a set of visual primitives and instructions for interactions and spatial manipulations of primitives.

Keywords: Natural language processing and understanding, semantic analysis, text-to-scene translation, visualization.

1 Introduction

Ongoing developments in the digital world poses a number of challenges. One of them is the increasing degree of text complexity. The user is exposed to more and more information in textual form for which advanced assistance techniques, such as automatic illustration generation and visualization of the textual content, might help the users to better comprehend it.

This paper describes ongoing research in visualization of simple narrative texts. The main contribution of the paper is a deep semantic analysis of text provided by the natural language processing pipeline, which involves recent developments in temporal and spatial information extraction. These semantic annotations are used for visualizing the recognized actions and the corresponding actors generated in a 3D graphics prototyping environment, and the actors and actions are rendered as animations. The results of this work that comprises manual and semi-automatic story animations are made available.

2 Related Work

In the last decade there has been a substantial amount of work in the domain of text-to-scene translation. The WordsEye project [1] focuses on an automated approach to generation of scenes from natural language. Johansson et al. [2] describe an automatic text-to-scene conversion system for the traffic accident

domain for Swedish. A similar work for French is also known [3]. Joshi et al. [4] propose an automatic text illustration system which automatically extracts keywords from text and relevant pictures are found in a database. Mihalcea and Leong [5] present a system for the automatic construction of pictorial representations for simple sentences. Zhu et al. [6] propose a Text-to-Picture system that generate a picture from a natural language text. Bui et al. [7] develop an application for automated translation of natural language patient instructions into pictures.

Our work is different to all the previous works in many aspects. First, we target the visualization task by producing animations and not static pictorial illustrations. Second, in contrast to the previous work, we do not rely on some sort of controlled-language descriptions, we use natural language texts. And third, we employ a number of wide-range semantic annotations extracted from text, including temporal and spatial annotations.

3 Natural Language Processing Pipeline

The visualization procedure that receives an input narrative and provides a visual representation of actions and involved actors relies on a natural language processing pipeline. This pipeline makes use of a number of automated syntactic and semantic analyses produced by external processing tools (for syntactic processing) as well as domain-specific algorithms and recognition models. Let us consider the following narrative:

Two Travellers were on the road together, when a Bear suddenly appeared on the scene. Before he observed them, one made for a tree at the side of the road, and climbed up into the branches and hid there. The other was not so nimble as his companion; and, as he could not escape, he threw himself on the ground and pretended to be dead. The Bear came up and sniffed all round him, but he kept perfectly still and held his breath: for they say that a bear will not touch a dead body. The Bear took him for a corpse, and went away. When the coast was clear, the Traveller in the tree came down, and asked the other what it was the Bear had whispered to him when he put his mouth to his ear. The other replied, "He told me never again to travel with a friend who deserts you at the first sign of danger."

Fig. 1. A narrative story from the Aesop’s fable corpus.

3.1 Event Recognition

In the NLP domain there are a number of definitions that regard events. In the Automatic Content Extraction context, events were defined as a complex structure with arguments. Each ACE event relates a predefined searchable topic of interest with arguments and a set of argument roles. A more honed definition of events was presented by Filatova and Hatzivassiloglou [8] which addressed some semantic representation for interpretations of “Who did what to whom when and where?” and was adopted in the novel annotation standard for temporal events, temporal expressions and temporal relations, and proposed in the TimeML markup annotation language [9]. In this work we also follow the TimeML-based definition of events as the most relevant representation of actions and their visualizations¹. Recent annotation efforts have shown that a simplified

¹ In this work we refer to actions as to a subset of temporally-anchored events as physical involvements of agents (protagonists) and interactions between them.

set of TimeML-based annotation guidelines has been found more practical and resulted in a higher interannotator agreement [10]. Moreover, such an annotation approach has given rise to sophisticated structured-output algorithms for temporal information processing [11].

3.2 Entity Recognition

Annotation of people, locations and other named entities for natural language processing is typically traced back to the Message Understanding Conferences (MUC), where the named entity task included identifying proper nouns in a text that referred to organizations, persons and locations [12]. In this work we primarily focus on the recognition of persons and locations as the central entities for action visualizations.

3.3 Event Participants Roles Recognition

Links from entities to the events they participate in have been annotated in a variety of different forms ([13–15]). In the scope of the described research with the goal of visualization of actions and actors, we primarily focus on action actors that are identified at the positions of syntactic subjects and objects for verb-triggered events. This information can be provided by dependency parsers [16] or derived from the syntactic constituent parse [17].

3.4 Coreference Resolution

Two words or phrases in a text are said to co-refer if they are both references the same object or person mentioned in the discourse. Discourse objects are mapped by the reader into elements of a mental model, that may or may not correspond to actual entities or events of the real world. In the full narrative discourse, co-referential links would form coreference chains where all mentions of a chain refer to the same entity. In computational linguistics a number of approaches to coreference resolution have been proposed. One of the most successful approaches with a high performance level is described in [18].

3.5 Temporal Ordering of Events

Narratives are a kind of texts that have a story plot and exhibit a particular structure of temporally ordered events, that is, a connected graph with nodes as events and edges as relations between the events labeled with tags such as *before* or *after* with respect to the temporal order in which the events occur. With the advent of TimeML, three major concepts of temporal information in text were defined: events, timexes (times) and temporal relations. Being thoroughly studied in the domain of newswire processing (e.g., [19]), temporal annotations of narrative timelines have been recently investigated in [11], which can derive temporal structures for narratives in terms of dependency relations that constitute a timeline, as presented in Fig. 2. In our work we employ this method.

3.6 From Events to Actions

So far we have described the tasks of semantic information extraction and natural language processing that are required for an automated “machine-based”

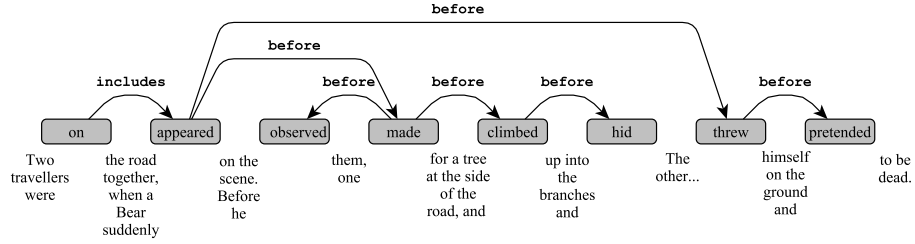


Fig. 2. A timeline for the narrative in Fig. 1. Nodes are events and edges are temporal relations signaled by linguistic cues in the text.

language understanding system. For visualization purposes however, not all recognized events are relevant or “visualizable”. Let us examine semantic classes of events. With respect to semantic classification of events, TimeML defines seven categories: reporting, perception, aspectual, intensional actions and states, states and occurrences. As Bethard et al. pointed out [10], human-based annotation of events and their semantic classes is a very difficult task even for experts in TimeML, and thus, they introduced the notion of *factual events* and *states*. Let us examine the events from the example presented in Fig. 1. One can distinguish the following (non-disjunctive) groups of events:

- factual events: *were* **on** the road ..., *suddenly* **appeared** on the scene ..., etc.
- events in possible worlds: *he could not* **escape**..., *a bear will not* **touch** ...
- states: *was not so* **nimble** as ..., *the coast was* **clear**, ...
- motions: *suddenly* **appeared** on the scene ..., *the bear* **came up** ...

As the first three groups are the instances of the TimeML semantic event classification scheme, motions are a quite novel class of events that recently has triggered a specific research interest. The most recent advances in this field are the SpatioTemporal Markup Language (STML) [20], and the Motion Markup Language (MotionML) [21]. With the similar objective, STML and MotionML were designed for capturing motions in text. While STML focuses on annotating motions with fine-grained semantics of spatial entities and their roles for further qualitative spatial reasoning, MotionML proposes a shallow annotation approach, where *trajectory*’s motions are annotated along with a single spatial label *path* triggered by *motion indicators*. The simplified motion annotation approach comes at the expense of the spatial semantic granularity, however, such an approach does not restrict the expressiveness of the annotation language and its potential scalability to STML. MotionML was proposed as an extension for the set of spatial roles introduced by the Spatial Role Labeling schema, and successfully adapted to the Spatial Role Labeling task [22]. For visualization purposes, we only focus on motion indicators and use them as a dictionary of visual actions. Fig. 3 represents a spatio-temporal structure of the narrative.

4 The Prototyping Environment Alice

Alice² is a 3D graphics prototyping environment developed for teaching and extending computational thinking for students of different age groups and back-

² <http://www.alice.org>

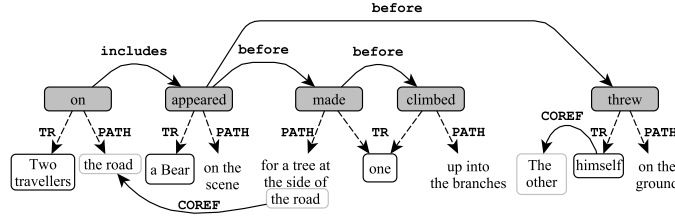


Fig. 3. A story timeline for the narrative in Fig. 1 with events considered for actions. Every action represents a motion in the story with argument roles such as *trajector* (TR) and *path* (PATH). Coreferential links are labeled with (COREF).

grounds. It provides a number of animated and non-animated graphical primitives (persons, animals, characters, vehicles, scenery objects, etc.), which can be placed in a virtual world with customized properties such as color, position opacity, and size. Let us summarize a list of procedural methods applicable to an object in Alice: (i) conversational procedures (**say** and **think**); (ii) orientations (**turn**, **roll**, **turnToFace**, etc.); (iii) positions (**move**, **moveToward**, **moveAwayFrom**, etc.); (iv) appearance (**setOpacity** and **setPaint**). Yet, Alice provides a number of object functional methods, i.e. the methods which return the object’s property value (similar to Java getters), for example: **getWidth()**, **getPaint()**, but also spatial properties, such as **isFacing(obj)**, **getDistanceTo(obj)**.

Since animated objects are internally represented as one of the skeletal joint systems, orientation procedures can also be applied to the parts of the skeletal joint system to model. Yet, Alice provides a set of programming statements such as **do_in_order**, **do_together**, **while**, **for_each** and allows one to create visualization/interaction scripts for visualizing complex events and interactions composed by atomic Alice instructions with respect to the user-defined logic.

5 Mapping Annotations to Graphical Primitives

After actions and actors in the text have been recognized by the NLP pipeline, we use these annotations to populate the virtual world in Alice. A number of assumptions, however, are made: (i) the provided annotations are disambiguated in terms or their meaning, i.e., they are mapped to a single synset in the lexical resource WordNet [23]; (ii) we manually determine visual appearance of characters in Alice; (iii) the spatial setting of the scene is predetermined. The initial spatial layout of the scene is presented in Fig. 4. After the spatial scene has been initialized, we generate Alice procedures from the annotations of actors and actions. The procedure is based on the mapping actions (and their actors) to a set of motion and conversational procedures. The lexical diversity of actions in text is treated by defining the root concepts in WordNet (**give voice**, **formulate#3**, and **state**, **say#1** for *saying*, and **travel**, **go#1**), so that if an action annotation cannot be directly matched with either a move or a say action, a search algorithm checks if the lexical item can be found in the list of synonyms for root actions or the root action is an inherit hypernym for the action in question. Each action receives the actor and a number of additional parameters, such as strings for conversational procedures, or absolute/relative directions and distances for



Fig. 4. The initial story setting with three actors (a Cat and a Hare as Travelers, and a Panda as the Bear). The palm tree represents *a tree at a side of the road*.

motions. Let us look at the instruction for the actions of a traveler (the cat) moving to the tree and climbing it:

```
\\the cat is moving to the tree
this.cheshireCat.moveToToward(
    this.coconutPalm, \\motion path, target
    this.cheshireCat.getDistanceTo(this.coconutPalm)); \\distance
\\the cat is climbing the tree
this.cheshireCat.move(
    MoveDirection.UP, \\motion direction
    this.coconutPalm.getHeight()*0.75); \\motion distance
where this.cheshireCat is the (actor) object and this.coconutPalm
is the path (target) of the motion.
```

Once visual instructions in Alice have been generated, temporal relations between events provided by the NLP pipeline specify the temporal order for execution of the instructions. As the recognized temporal relation labels are merely text strings, they have to be transformed into some computational form. One of the computational forms for representing temporal relations is the Allen’s temporal interval algebra. We first select actions which are in `includes` or `is_included` relations to each other and generate a `do_together` instruction block that includes them. Temporal relations with the label `identity` signal merging temporal events to their antecedents. After that, other temporal relations with labels `before` and `after` are inverted to a single label `after` for generating a linear order of instruction execution in Alice. Two animations are available³: (i) manually created, and (ii) generated semi-automatically.

6 Discussion and Conclusions

In this paper we have presented the first prototype for narrative visualizations. The technique employs a natural language processing pipeline for syntactic and semantic analysis of text. At the semantic level it provides information about actions and actors, their relations, temporal ordering of the actions on the timeline, and spatial roles for the identified actions. The visualization technique translates

³ anonymized

semantic annotations into instructions in the visualization environment by mapping these annotations into visual primitives with a number of assumptions. We recognize these assumptions as bottlenecks for a generic and fully domain-independent visualization procedure, and would like to provide the reader with the emerging directions that address these challenges.

Temporal ordering is essential for text comprehension and received great interest in the research community. In language, temporal ordering is implemented by aspectual and tensed cues. However, in many cases this information is not enough to provide an accurate temporal analysis since most of the cues reside in commonsense knowledge. *Event durations* are very important information for visualization, however, there is very little evidence in text about them. This information humans receive from other sources (e.g. personal experience and observations). The lack of duration information prevents us from exact visual replication of the story plot (and also using TimeGraph [24] as the computational means for timelines), however, for the chosen text genre and text complexity, the visualized story reached a good approximation level of the plot semantics.

Spatial information is a type of semantic annotations vital for visualizations. All visualizations techniques are designed around motions. At the same time spatial information is always present in narratives, but an automated spatial analysis of text remains a far reaching goal. Two important research initiatives addressed spatial annotations in text: STML and MotionML. In this work we used MotionML and the corresponding annotated corpus for identifying motion actions in text. STML, on the other hand, is designed for annotating fine-grained spatial semantics, and seems to be more powerful for consequent qualitative spatial reasoning, but, at this moment, no annotations in STML are available. Spatial annotations are very important for visualizing the initial spatial scene, but similarly to temporal ordering, this information is usually not directly available in text, where personal commonsense knowledge primes this visualization.

Acknowledgments

The presented research was supported by the TERENCE (EU FP7-257410) and MUSE (EU FP7-296703) projects.

References

1. Coyne, B., Sproat, R.: WordsEye: an automatic text-to-scene conversion system. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM (2001) 487–496
2. Johansson, R., Berglund, A., Danielsson, M., Nugues, P.: Automatic text-to-scene conversion in the traffic accident domain. In: Proceedings of the International Joint Conference on Artificial Intelligence. (2005) 1073–1078
3. Kayser, D., Nouioua, F.: From the textual description of an accident to its causes. *Artificial Intelligence* **173**(12) (2009) 1154–1193
4. Joshi, D., Wang, J.Z., Li, J.: The story picturing engine—a system for automatic text illustration. *ACM Transactions on Multimedia Computing, Communications, and Applications* **2**(1) (2006) 68–89
5. Mihalcea, R., Leong, C.W.: Toward communicating simple sentences using pictorial representations. *Machine Translation* **22**(3) (2008) 153–173

6. Zhu, X., Goldberg, A.B., Eldawy, M., Dyer, C.R., Strock, B.: A text-to-picture synthesis system for augmenting communication. In: Association for the Advancement of Artificial Intelligence. (2007) 1590–1595
7. Bui, D., Nakamura, C., Bray, B.E., Zeng-Treitler, Q.: Automated illustration of patients instructions. In: Proceedings of the Annual Symposium of the American Medical Informatics Association, AMIA (2012) 1158–1167
8. Filatova, E., Hatzivassiloglou, V.: Domain-independent detection, extraction, and labeling of atomic events. In: Proceedings of the Conference on Recent Advances in Natural Language Processing. (2003)
9. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G.: TimeML: Robust specification of event and temporal expressions in text. In: Proceedings of the International Workshop on Computational Semantics. (2003)
10. Bethard, S., Kolomiyets, O., Moens, M.F.: Annotating narrative timelines as temporal dependency structures. In: Proceedings of the International Conference on Linguistic Resources and Evaluation. (2012)
11. Kolomiyets, O., Bethard, S., Moens, M.F.: Extracting narrative timelines as temporal dependency structures. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, ACL (2012) 88–97
12. Chinchor, N., Robinson, P.: MUC-7 Named entity task definition. In: Proceedings of the 7th Conference on Message Understanding. (1997)
13. Marcus, M.P., Marcinkiewicz, M.A., Santorini, B.: Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics* **19**(2) (1993)
14. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The Berkeley FrameNet project. In: Proceedings of the 17th international conference on Computational linguistics, ACL (1998) 86–90
15. Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics* **31**(1) (2005) 71–106
16. Nivre, J., Hall, J., Nilsson, J.: Maltparser: A data-driven parser-generator for dependency parsing. In: Proceedings of LREC. (2006) 2216–2219
17. De Marneffe, M.C., Manning, C.D.: The Stanford typed dependencies representation. In: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation, ACL (2008) 1–8
18. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford’s multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, ACL (2011) 28–34
19. Verhagen, M., Saurí, R., Caselli, T., Pustejovsky, J.: SemEval-2010 Task 13: TempEval-2. In: Proceedings of the 5th International Workshop on Semantic Evaluation. (2010) 57–62
20. Pustejovsky, J., Moszkowicz, J.L.: The qualitative spatial dynamics of motion in language. *Spatial Cognition & Computation* **11**(1) (2011) 15–44
21. Kolomiyets, O., Moens, M.F.: MotionML: A shallow approach for annotating motions in text. In: Proceedings of Corpus Linguistics. (2013)
22. Kolomiyets, O., Kordjamshidi, P., Moens, M.F., Bethard, S.: SemEval-2013 Task 3: Spatial Role Labeling. In: Proceedings of the Seventh International Workshop on Semantic Evaluations, ACL (2013) 255–262
23. Miller, G.A.: WordNet: a lexical database for English. *Communications of the ACM* **38** (1995) 39–41
24. Miller, S.A., Schubert, L.K.: Time revisited. *Computational Intelligence* **6**(2) (1990) 108–118